04 Hailstone Sequence

Take any natural number *n*. If *n* is even, divide it by 2 to get *n // 2*. If *n* is odd, multiply by 3 and add 1 to obtain *3 n + 1*. Repeat the process indefinitely. The conjecture is that no matter what number you start with, you will eventually reach 1.

This conjecture has never been proved. But it has been verified on the computer for all starting values up to 20 * 2^{58}. To disprove this conjecture one has to find a single starting number that goes into a cycle that does not contain 1.

```
The function can be written as follows:

     f ( n ) = ( n // 2 ) if n is even

     f ( n ) = ( 3 * n + 1 ) if n is odd
```

For a given starting number, this function generates a series of numbers ending at 1 that is called the hailstone sequence. The hailstone sequences for starting numbers 7, 8, and 9 are:

```
7 22 11 34 17 52 26 13 40 20 10 5 16 8 4 2 1

8 4 2 1

9 28 14 7 22 11 34 17 52 26 13 40 20 10 5 16 8 4 2 1
```

All sequences tested so far end in an endless cycle of 4, 2, and 1. Hence the sequences are halted when they reach 1. We will define the cycle length of a sequence to be the number of steps it takes from the starting number to reach 1. Here are the cycle lengths of a few starting numbers:

Number	Cycle Length
1	0
2	1
3	7
4	2
5	5

In your program you will verify this conjecture in a user defined range. You will prompt the user to enter the starting number of the range and the ending number. After you have read **both** the numbers, you will check if both numbers are positive (> 0) individually and then you will check that the starting number in the user defined range is strictly less than the ending number in that range. If the input does not meet the specifications, prompt the user to enter both the starting and ending numbers.

Once the beginning and ending of the range have been verified, your program will compute the cycle length for each of the numbers in that range inclusive of the end points. Your program will print out the number that has the largest cycle length and what that cycle length is. If there are two or more numbers that have the same largest cycle length, then print only the last number (greatest number) that has the largest cycle length.

Your sample session will look like this:

```
Enter starting number of the range: 1

Enter ending number of the range: 5

The number 3 has the longest cycle length of 7.
```

In your program all your computations will be in the function main(). You must use nested loops to obtain your results.